Journal of Dynamics, Monitoring and Diagnostics, 2025, 4, 250-263
https://doi.org/10.37965/jdmd.2025.795

Fault Detection in Wind Turbine Bearings by Coupling Knowledge Graph and Machine Learning Approach

Paras Garg,[1] Arvind Keprate,[2] Gunjan Soni,[1] A.P.S. Rathore,[1] and O.P. Yadav[3]

[1]Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur, Jaipur, India
[2]Green Energy Lab, Department of Mechanical, Electrical and Chemical Engineering,
Oslo Metropolitan University, Oslo, Norway
[3]Department of Industrial and Systems Engineering, North Carolina Agricultural and
Technical State University, Greensboro, NC, USA

(Received 22 April 2025; Revised 15 July 2025; Accepted 04 September 2025; Published online 09 September 2025)

Abstract: Fault sensing in wind turbine (WT) generator bearings is essential for ensuring reliability and holding down maintenance costs. Feeding raw sensor data to machine learning (ML) model often overlooks the enveloping interdependencies between system elements. This study proposes a new hybrid method that combines the domain knowledge via knowledge graphs (KGs) and the traditional feature-based data. Incorporation of contextual relationships through construction of graph embedding methods, such as Node2Vec, can capture meaningful information, such as the relationships among key parameters (e.g. wind speed, rotor Revolutions Per Minute (RPM), and temperature) in the enriched feature representations. These node embeddings, when augmented with the original data, can be used to allow the model to learn and generalize better. As shown in results achieved on experimental data, the augmented ML model (with KG) is much better at predicting with the help of accuracy and error measure compared to traditional ML methods. Paired t-test analysis proves the statistical validity of this improvement. Moreover, graph-based feature importance increases the interpretability of the model and helps to uncover the structurally significant variables that are otherwise ignored by the common methods. The approach provides an excellent, knowledge-guided manner through which intelligent fault detection can be executed on WT systems.

Keywords: anomaly detection; knowledge graph embedding; machine learning; wind turbine fault detection

I. INTRODUCTION

Wind energy has emerged as a thriving renewable energy source, with maintenance playing a crucial role in wind turbine (WT) performance [1] and cost-effectiveness. Analyzing failure modes, causes, and crucial component identification techniques is part of WT maintenance [2]. Power system reliability may be adversely affected by wind power's unpredictable nature, which calls for a variety of evaluation techniques [3]. Significant differences exist in failure rates and downtimes among WT subassemblies, according to reliability data gathered from several databases; in general, offshore WTs have greater failure rates than onshore ones [4]. Reducing lifecycle costs and increasing power generation efficiency require improved component reliability [5]. There is a significant, nonlinear correlation between WT reliability and annual energy production as well as operation and maintenance costs, underscoring the significance of determining the best reliability enhancements to reduce the levelized cost of energy [4]. The health of gas turbine and WT engines depends heavily on bearings, and when they fail, there is a substantial downtime and maintenance expense [6]. Uneven operating stress and climatic circumstances present special problems for WT bearings, requiring improvements in surface engineering, design, and lubrication [7]. Prognostics and health management algorithms have been developed for WT engines in order to identify early signs of critical bearing failure. These methods determine the remaining usable life by combining material-level fatigue models with vibration transducer sensor data [8]. Through the use of model-based estimates in the absence of diagnostic indicators and monitored features like vibration and oil debris at later stages, the combination of health monitoring data and model-based techniques offers a comprehensive prognostic capability throughout a bearing's life [9].

The application of machine learning (ML) techniques to improve WT predictive maintenance has shown great promise. Numerous strategies have been investigated, such as hybrid models that combine ML algorithms and statistical process control [10] and data-centric techniques. With decision tree and XGBoost models reaching over 90% accuracy, these techniques have proven to be highly accurate in problem diagnosis and maintenance prediction [11]. Support vector machines and convolutional neural networks have been used to estimate long-cycle maintenance times [12]. Autoencoders outperform other algorithms in detecting operational anomalies, and recent developments include the creation of entire frameworks, including anomaly detection and prognostics. Furthermore, Long Short-Term Memory (LSTM) neural networks have demonstrated potential in forecasting the essential components' remaining useful lives [13]. These ML-based strategies help wind farm operations run more efficiently overall by reducing downtime and optimizing maintenance schedules. Conventional ML models are effective at recognizing patterns, but they ignore the inherent relationships between their input

Corresponding author: Arvind Keprate (e-mail: arvind.keprate@oslomet.no)

data [13] and instead treat them as separate independent variables. Temperature, vibration, and rotor speed are some of the parameters that affect each other under operating force and climatic conditions in the WT system, particularly in its generator bearings. This limitation limits the interpretability and accuracy of ML-based defect identification. Knowledge graphs (KGs) are a useful tool for leveraging subject expertise and using semantic relationships to explicitly model complex interdependence [14]. When KG-based embeddings and sensor data are combined, ML models' contextual awareness is improved, leading to more powerful but explicable predictive capabilities. In order to improve accuracy and insight in identifying problems in WT generator bearings, this study presents the creation of a hybrid fault detection model that combines KGs and ML.

To address existing limitations, this research creates a new hybrid framework that combines domain-specific knowledge structures contained in KGs with traditional feature data in order to improve WT generator fault detection. This hybrid framework targets existing model limitations by establishing better connections between system parameters with more effectiveness.

The primary objective of this paper is as follows:

1. To create an advanced ML framework that unites Node2Vec node embeddings from KG techniques with classical sensor data for precise WT generator bearing fault detection.
2. To assess the performance elevation of KG + ML methodology relative to standard ML approaches by using statistical significance testing along with quantitative measurement methods.

The remainder of this paper is structured as follows: Section II presents the literature review on WT maintenance. Section III illustrates how the KG was integrated with ML model using node embedding technique. Section IV discusses the experimental results and performance evaluation of the hybrid model. Finally, Section V concludes the study and outlines directions for future research.

II. LITERATURE REVIEW

Efficient renewable energy production by WTs demands advanced automatic failure detection systems that maintain system quality along with operational effectiveness. Generator bearings in WT experience high rates of failure because they face ongoing mechanical forces, unstable loads, and environmental conditions [1]. The detection of early faults in these bearings requires immediate attention because it prevents major system failures while reducing maintenance expenses and system downtime. Fault detection through traditional methods depends on sensor readings from vibration, temperature, and acoustic measurements that conventional ML algorithms examine. The detection techniques employ Support Vector Machine (SVM) along with K-Nearest Neighbours (KNN) and DT algorithms, and Random Forest (RF) models as their main execution tools. [15] developed a system by applying SVM to vibration signal time-domain statistical features in order to detect faults in WT gearboxes and bearings. The study presented by [16] showcased RF classifiers as effective tools for abnormal pattern detection in rotating machinery. The performance of existing approaches is acceptable, but they offer restricted interpretability when analyzing interdependencies among parameters in WT systems. The recent developments in artificial intelligence, along with semantic technologies, have brought KGs as an advanced framework to structure domain knowledge representation. KGs enable semantic networks that establish associations between components, parameters, and failure modes, thereby making context-based choices possible. [17] showed how KG enables effective applications in industrial systems, especially for equipment maintenance and health monitoring. Many industrial sectors, including healthcare [18], e-commerce [19], and cybersecurity [20], have adopted KGs to enhance ML models together with domain knowledge and explainability features.

The use of KGs as a WT fault detection approach shows minimal current application despite their advantages. The integration of KGs with ML models creates an effective challenge for practical applications. Embedding techniques transform KGs into low-dimensional vector spaces according to recent academic research. Node2Vec [21] serves as a scalable approach that maps graph nodes into latent spaces through random walk computations while maintaining node structural patterns. The embeddings become compatible with sensor-based features, which allows ML models to gain knowledge about statistical patterns and semantic relationships simultaneously. Researchers have conducted limited studies regarding the application of graph-based techniques for predictive maintenance in renewable energy systems. The authors [22] presented a hybrid graph neural network (GNN) model, which integrated time-series data and structural component interactions for WT fault prediction systems. The researchers discovered that including turbine subsystem relational data increased the precision rate of fault detection. Their methodology needed extensive domain expertise during graph building, while the computations remained too complex. While GNN-based approaches are outstanding among the best ranking models, the investigated integration employing Node2Vec is lighter and more flexible. Such an approach is employed within this research to build a domain-specific KG that depicts dependencies within important WT parameters like rotor speed, bearing temperature, wind speed, and generator torque, among others, and to incorporate this graph in feature vectors. These embedded vectors are then incorporated with normal sensor data for training ML models like RFs to obtain enhanced performance during fault detection. Also, disadvantages of the combined approaches, which have been incorporated in this study, include better model interpretability. Since knowledge captured in the KG would have been derived based on expert knowledge in the respective field, the practitioner would have an understanding of why the model came up with certain predictions in areas such as WT maintenance. Further, employing KGs can improve data augmentation, allow handling of missing values better, and the need for great deal of labeled data diminishes as well. In conclusion, while the traditional set of ML models has been successfully applied to the WT fault detection problem, the incorporation of KBR approach is a step further that allows the representation of system complexity. The approach of combining KG-based embeddings with the traditional ML features that were discussed in this study can allow for developing more accurate and explainable models for the early fault diagnosis in WT generator bearings.

III. PROPOSED FAULT DETECTION FRAMEWORK

A. KNOWLEDGE GRAPH

KG presents structured information through nodes and edges for representing entities as nodes and their connections or relations as edges [23]. Real-world entities become nodes in the graph structure, whereas edges depict the relations between them. KG implements semantic triples as a data structure for machine-readable representation of factual information through subject, predicate, and object relations. The relationship between turbines and components emerges through the triple entry (wind speed, affects, and generator bearing temperature). The power of KGs stems from their capability to bring together diverse heterogeneous data sources into one unified model, which enhances search intelligence and discovery capability as well as reasoning abilities [24]. The wide range of applications includes natural language processing as well as recommender systems and information retrieval. The graph structure of data storage enables KGs to accept multiple flexible queries for pattern detection using analytics tools such as centrality clustering and path analysis. The visual format, along with interconnected structure of graph databases, makes them user-friendly for studying complex datasets together with knowledge domains [25]. Data storage systems gain semantic abilities through KGs because these systems allow machines to understand both contents and patterns between data points.

The mathematical definition of graph is a structure made of vertices and edges.
Where,

- V: a set of vertices;
- E: a set of edges;
- V = {v1, v2, v3, v4, v5, v6} (entities or nodes);
- E = {(v1, v5), (v1, v3), (v6, v3), (v4, v5)} (edges or relations).

Fig. 1 illustrates a KG, which represents entities (shown as nodes) and their relationships (shown as edges). Each node corresponds to a real-world concept (e.g., *Person*, *Country*, *and City*), and each directed edge defines a specific relationship between entities (e.g., *born in*, *located in*, *and works for*).

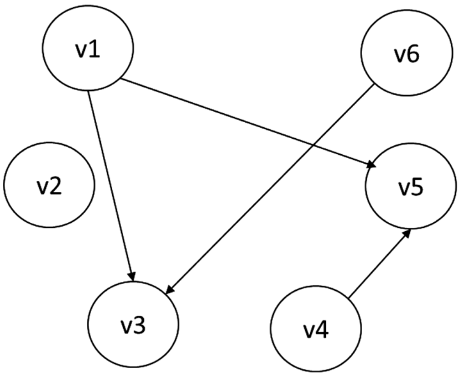


Fig. 1. Visual representation of KG.

B. COUPLING KG AND ML

The proposed hybrid fault detection framework, as illustrated in Fig. 2, shows how the proposed KG + ML pipeline operates for detecting WT faults. Sensor data procurement leads to the construction of a KG that displays system component associations and dependencies. Extracted node centrality measures from this KG that get combined with regular sensor data features to form improved data. The ML model uses an enriched data set that results from this process in order to undergo training and testing. A trained model creates predictive outcomes before standard performance criteria measure their accuracy. The feedback loop from evaluation enables permanent advancements to be made in the model's performance. The framework improves both predictive ability as well as interpretability through its integration of domain knowledge through the KG to provide a strong and explainable solution for bearing fault detection in WTs. The proposed framework combines conventional Supervisory Control and Data Acquisition (SCADA)-based data processing with semantic knowledge embedding using KG to enhance the performance of ML models in detecting generator bearing faults. ML applications benefit from KGs through their capability to enhance raw data with meaning-based structures, together with improved feature-building abilities, which enable models to predict with superior information [26]. Stand-alone ML models become more powerful when coupled with KGs because the combination improves interpretability and enhances data integration and performance, particularly in recommendation systems and predictive maintenance and supply chain optimization environments. KGs acquire their status as vital components of next-generation intelligent systems through their flexible query capabilities, which emerge from their graph-based structure. The study demonstrates domain-specific learning benefits when KG merges with ML because the performance metrics showed enhanced improvement.

Fig. 2 shows the illustration of the end-to-end process of proposed hybrid framework of fault detection of WT generator bearings, which has been arranged into two significant parts:

1. Data Preparation and KG Construction: The first stage of it is the gathering of the SCADA sensor data for the features like rotor speed, bearing temperature, and wind speed. The data are then pre-processed and cleaned to maintain time synchronicity, uniformity, and quality. To determine valuable interdependencies between features, correlation analysis was carried out. This knowledge, together with the knowledge acquired by professionals in the realm, is utilized to create a KG in which nodes represent the important features of SCADA (e.g., Wind Speed, Gen_RPM). And edges represent the functional or operational dependencies among the features.
2. ML Integration and model training: After the KG construction, Node2Vec algorithm was used to produce low-dimensional node embeddings. These embeddings contain structural and relational characteristics of the graph. The derived graph-based vectors are then merged with the basic SCADA feature data to provide an augmented feature data. This augmented dataset was used to train a ML model (e.g., XGBoost, RF etc.) to predict faults. Standard measures are used to assess model performance, e.g. mean absolute error

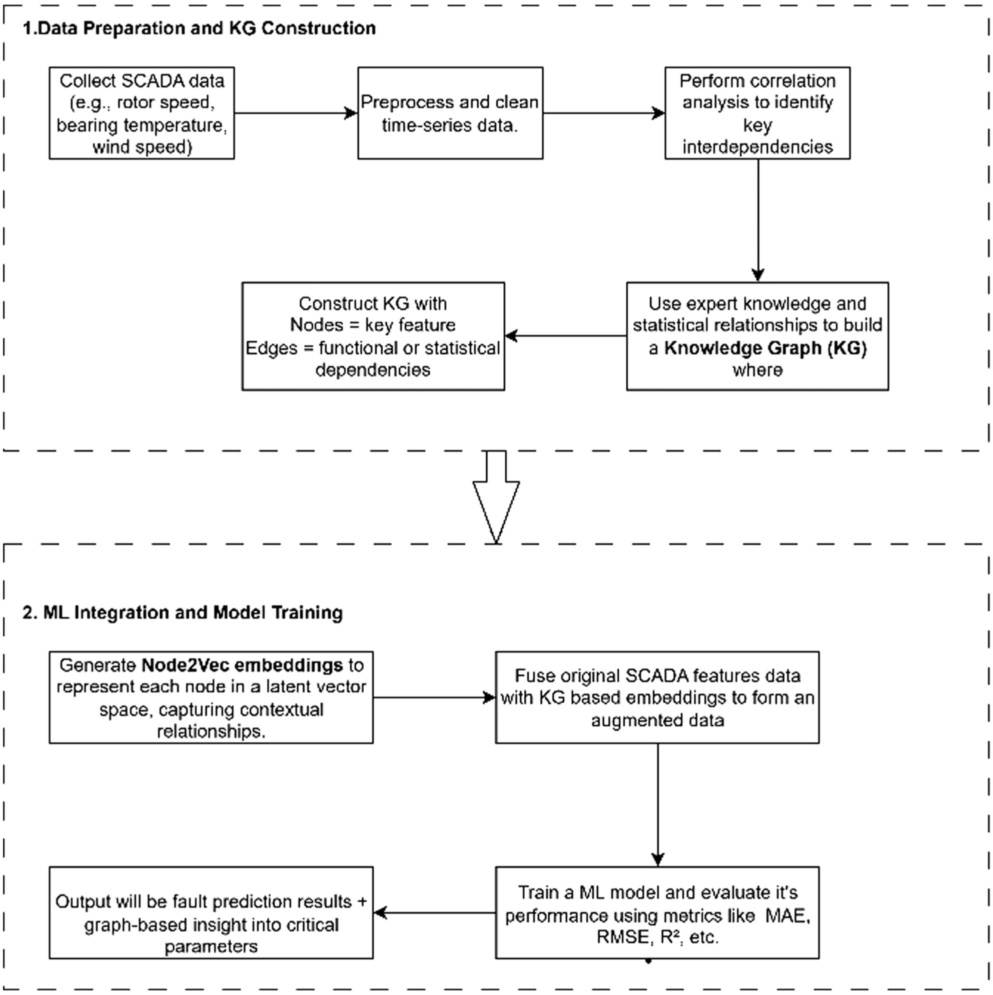

1.Data Preparation and KG Construction
Collect SCADA data (e.g., rotor speed, bearing temperature, wind speed)
Preprocess and clean time-series data.
Perform correlation analysis to identify key interdependencies
Use expert knowledge and statistical relationships to build a Knowledge Graph (KG) where
Construct KG with Nodes = key feature Edges = functional or statistical dependencies
2. ML Integration and Model Training
Generate Node2Vec embeddings to represent each node in a latent vector space, capturing contextual relationships.
Fuse original SCADA features data with KG based embeddings to form an augmented data
Train a ML model and evaluate it's performance using metrics like MAE, RMSE, R², etc.
Output will be fault prediction results + graph-based insight into critical parameters

Fig. 2. Proposed hybrid fault detection framework combining SCADA-based feature data with knowledge graph embeddings.

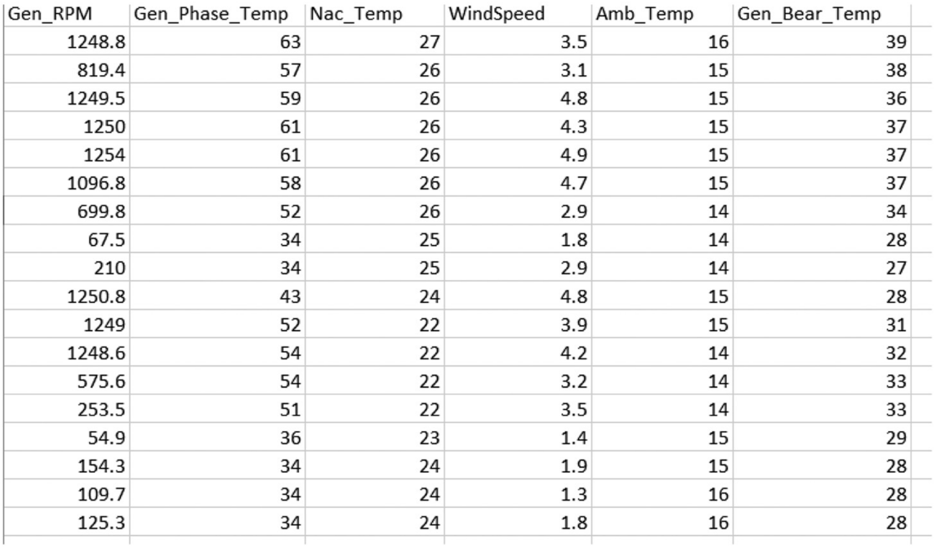
(MAE), root mean square error (RMSE), and R 2. The output is not only shown in the results of fault prediction but also graph-based interpretability information, which provides insights into structurally important parameters usually ignored by conventional models.

C. DATA COLLECTION

The research dataset comes from SCADA systems, which EDP (2017) provided. The dataset contains operational data, which were gathered from four horizontal-axis WTs installed along the western coastline of Africa. A two-year record of data exists from 2016 through 2017 with 10-minute averaged measurements that show complete turbine operations throughout the period. The total number of parameters measured amounts to 76, which provides detailed information about turbine performance as well as health status. The SCADA readings accompany meteorological data points that match the same timestamp, which helps explain environmental factors affecting turbine operation. Supervised ML applications rely heavily on failure logs, which contain timestamps together with descriptions about failed components as well as pertinent comments about their failed state. For this case, Turbine Number 7 (“T07”) was adopted because its failure log indicates a generator bearing fault, which is an area of interest in this research. The total numbers recorded for T07 are 52445 in the year 2016 and 52294 in the year 2017. With such instances being available for training as well as testing the fault detection model, their utilization in the analysis is appropriate. A subset of relevant features was selected from the full SCADA dataset (as shown in Fig. 3) for model development. These features, along with the target variable indicating fault or normal status, are summarized in Table I.

D. EXPLORATORY DATA ANALYSIS

Fig. 4 shows how five key operational parameters of a WT affect generator bearing temperature through box plots analysis, including Gen_Bearing_Temp. Gen_RPM, Gen_Phase_Temp, Nac_Temp, WindSpeed, and Amb_Temp. The analysis shows a linear positive relationship between the generator phase heat and the bearing temperature, thus indicating direct thermal connections between these two operational elements. As Gen_RPM increases together with WindSpeed, the bearing temperatures elevate because higher mechanical workload occurs during energy conversion [27]. The nacelle temperature impact on bearing temperatures is moderate since nacelle temperature elevation leads to higher temperatures but shows wider variation in changes. The connection between ambient temperature



Gen_RPM	Gen_Phase_Temp	Nac_Temp	WindSpeed	Amb_Temp	Gen_Bear_Temp
1248.8	63	27	3.5	16	39
819.4	57	26	3.1	15	38
1249.5	59	26	4.8	15	36
1250	61	26	4.3	15	37
1254	61	26	4.9	15	37
1096.8	58	26	4.7	15	37
699.8	52	26	2.9	14	34
67.5	34	25	1.8	14	28
210	34	25	2.9	14	27
1250.8	43	24	4.8	15	28
1249	52	22	3.9	15	31
1248.6	54	22	4.2	14	32
575.6	54	22	3.2	14	33
253.5	51	22	3.5	14	33
54.9	36	23	1.4	15	29
154.3	34	24	1.9	15	28
109.7	34	24	1.3	16	28
125.3	34	24	1.8	16	28

Fig. 3. Sample view of the input dataset featuring key variables.

Table I. Selected features and target for developing the model [27].

Variable	Description	Units
	Features	
Gen_RPM	Generator shaft/bearing rotational speed	rpm
Gen_Phase_Temp	SCADA dataset gives the average temperature inside generator in stator windings, Phases 1, 2, and 3. Since the temperatures are nearly the same, Gen_Phase_Temp is an average temperature of the three temperatures	°C
Wind_Speed	Ambient wind speed	m/s
Amb_Temp	Air ambient temperature	°C
Nac_Temp	Nacelle temperature	°C
Target		
Gen_Bear_Temp	Temperature in generator bearing 1 (Driven End)	°C

and bearing temperature appears weaker according to the results presented by Amb_Temp data points. The analysis demonstrates that thermal and mechanical stresses in WTs create a relationship where Gen_Phase_Temp proves to be the primary element for forecasting Gen_Bearing_Temp.

This study used correlation analysis to show how different SCADA indicators work together and which depend on one another to produce generator bearing faults. A graphical representation shows how strongly each pair of chosen features affects the other. Fig. 5 illustrates the correlation relationship between input features and target. The measured signals display significant associations with one another. Example: a) wind speed and generator rotational speed, b) wind speed and generator phase temperature, and c) generator phase temperature and bearing temperature. The matrix shows that the chosen features demonstrate their strong connection to the end measurement variable (target). The pairs of measurements need additional testing for their relationship with our results. The training set shows how target values relate to each other. The correlation map determines the connections between parameters that form the relationships of the KG. Going beyond standard feature vectors, the model will capture context-aware interactions that are frequently overlooked by typical ML techniques by including information from the correlation map into the KG.

E. KG CONSTRUCTION AND VISUALIZATION

KG is a structured representation of domain knowledge, where entities (such as turbine components or environmental conditions) are connected by relationships that describe their interactions or dependencies. In the proposed research, the KG is being built to capture both semantic and functional links between the most important SCADA parameters impacting the health of the generator bearing. The nodes relate to operational parameters, including the wind speed, generator RPM, nacelle temperature, ambient temperature, and generator phase temperature. The edge determination is done in a hybrid style where (i) statistical correlation (e.g. Pearson correlation coefficients) is used to pin out pairs of parameters that have a significant interaction and (ii) domain expert knowledge is used to verify and augment such relationships on the basis of known mechanical and thermal behavior of the various components of the WT structure. As an example, both wind speed and generator RPM are associated because the two are directly dependent on each other physically, whereas the nacelle temperature is associated with generator phase temperature because of the thermal influence on the latter. These links are further polished with failure logs and past trends. The combination of data-driven semantics and expert-defined knowledge guarantees that the KG will have the right statistics as well as operational dependencies. The resulting structure of KG is shown in Fig. 6 where the nodes involve SCADA features and the edges indicate either functional or statistically defined interdependencies. Each edge is labeled with the relationship type (“affects,” “correlates with,”, etc.) based on expert-defined heuristics. By encoding these

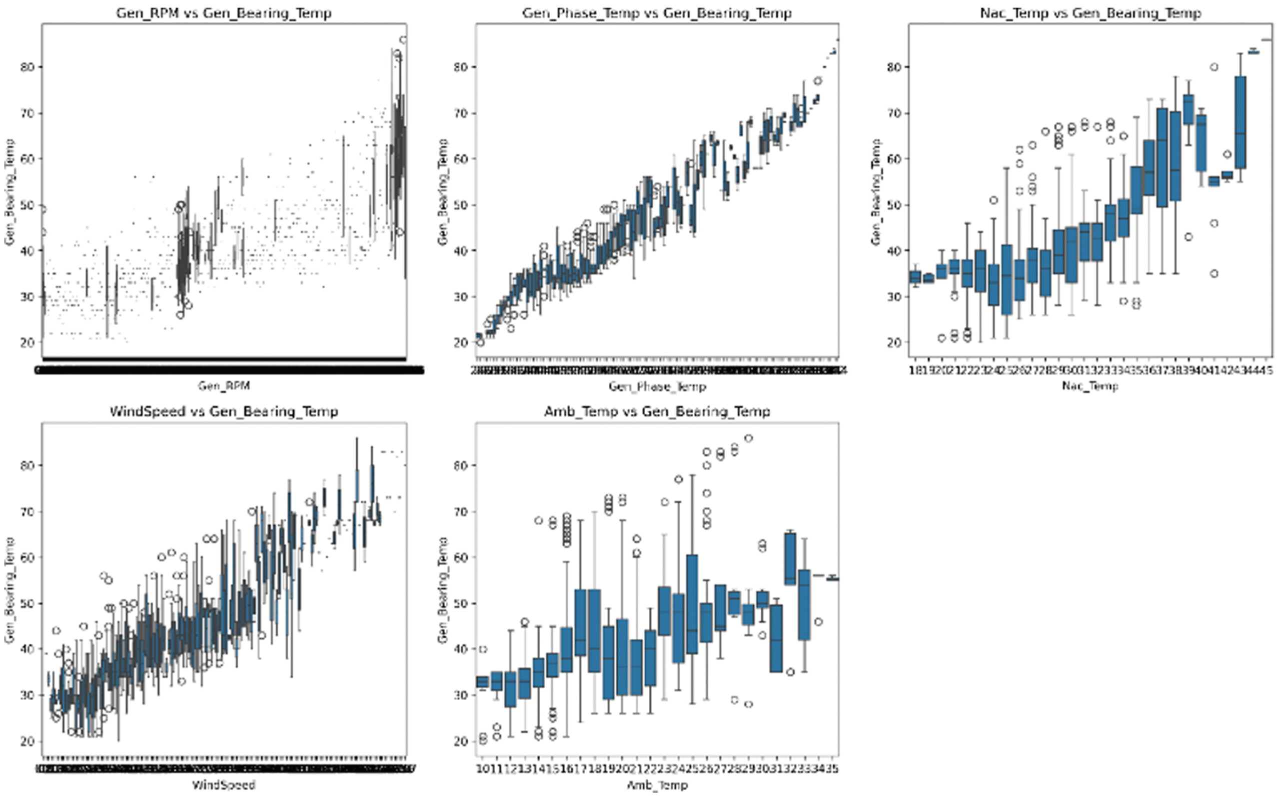


Fig. 4. Box plots showing the distribution of Gen_Bearing_Temp (Generator Bearing Temperature) with respect to key wind turbine parameters.

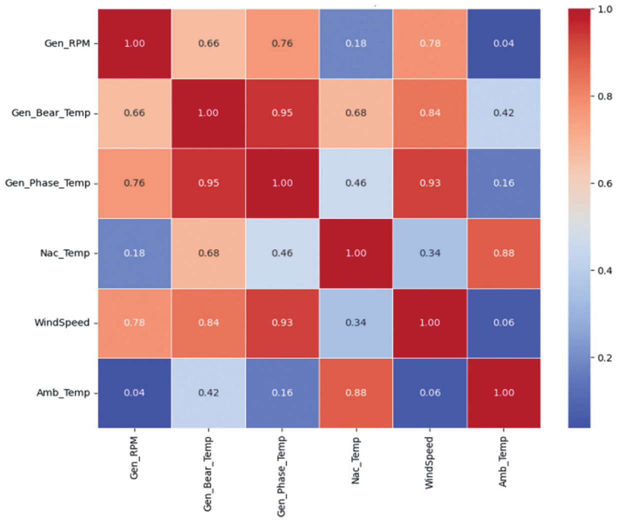


Fig. 5. Pearson correlation matrix of the input features.

interdependencies, the KG serves as a contextual layer that complements raw sensor data. To integrate this knowledge into the ML pipeline, the graph is transformed into low-dimensional vector representations using KG embedding techniques. These embeddings preserve the relational structure and allow the ML model to reason about the system more holistically, leading to improved fault detection accuracy and interpretability.

The study utilizes SCADA sensor data representing the operational states of a WT generator.

- Entities include: Wind Speed, Generator RPM, Nacelle Temperature, Ambient Temperature, Generator Bearing Temperature, and Generator Phase Temperature.
- Entity Relations: E.g.,
  - ⇒ Wind Speed, Generator RPM, Nacelle Temperature, Ambient Temperature, and Generator Phase Temperature ↔ Gen bearing temperature (all affects).
  - ⇒ Generator RPM ↔ Generator Phase Temperature (Direct correlation through mechanical and electrical load).
  - ⇒ Nacelle Temperature ↔ Generator Phase Temperature (Environmental influence on internal temperature).
  - ⇒ Wind Speed ↔ Generator RPM (Direct correlation through blade speed).
  - ⇒ Wind Speed ↔ Nacelle Temperature (Affects operational heat and cooling).
  - ⇒ Ambient Temperature ↔ Nacelle Temperature & Generator Phase Temperature (Affects cooling efficiency and risk of condensation).

F. GRAPH EMBEDDING AND AUGMENTED DATASET

The next stage is to translate the KG, which is used to model the interdependencies among SCADA parameters, into a numerical structure that ML algorithms can comprehend. This is accomplished by a procedure known as graph embedding, in which every node—which stands for a feature or parameter—is mapped to a low-dimensional vector that encapsulates the node's relational and structural

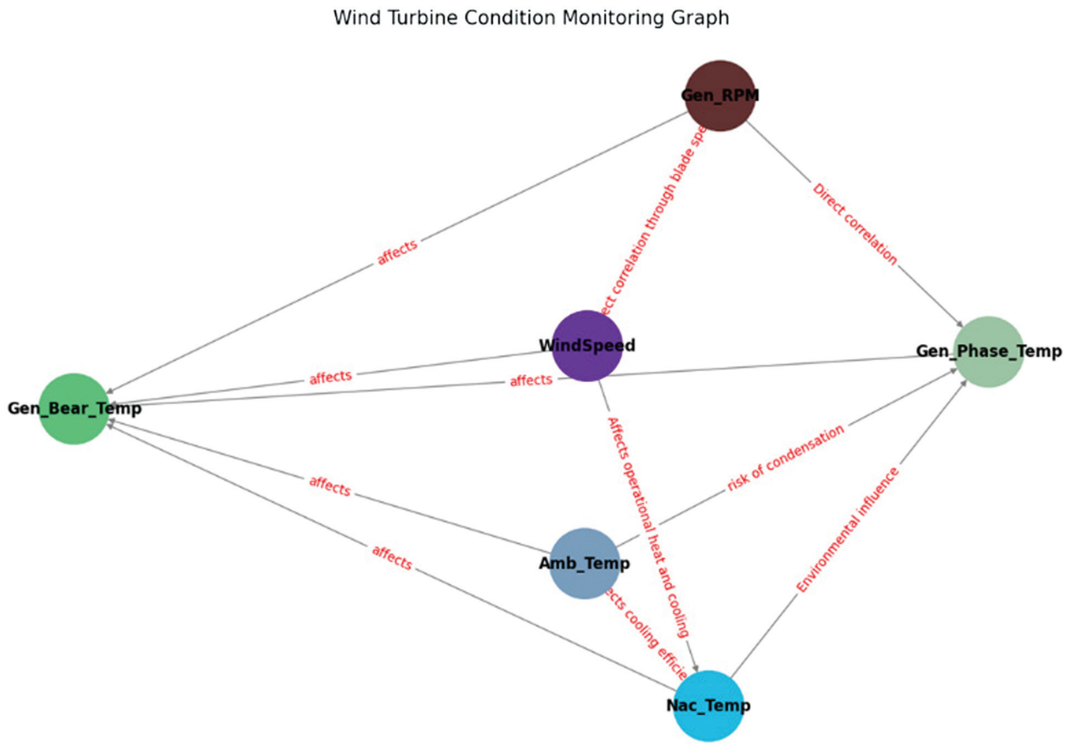


Fig. 6. Knowledge graph representation used for wind turbine condition monitoring. Each node corresponds to a SCADA parameter: Gen_RPM, Gen_Phase_Temp, Gen_Bear_Temp, WindSpeed, Amb_Temp, and Nac_Temp. Directed edges represent known functional or statistical relationships between features. Edge labels describe the nature of each dependency, e.g., “affects,” “environmental influence,” “risk of condensation,” or “direct correlation.” These relations were derived using a combination of expert knowledge and correlation analysis.

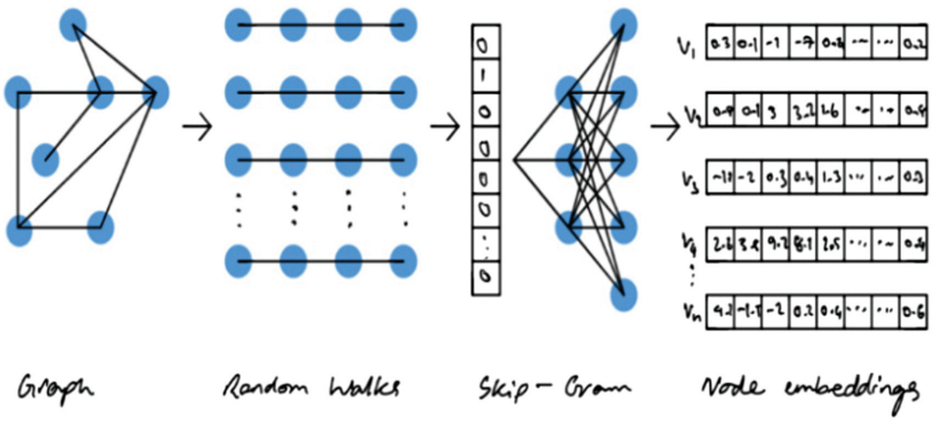


Fig. 7. Node2Vec Architecture (Towards Data Science, 2022).

context within the graph. A rich feature space for model training is created by combining these embeddings with the original SCADA data to create an enhanced dataset.

Through this integration, the model is able to deduce links, patterns, and impacts that are recorded in the domain knowledge network in addition to learning from the raw sensor values. Consequently, the ML model gains strength, interpretability, and the ability to recognize tiny indications of bearing failure.

1) NODE2VEC: EMBEDDING KNOWLEDGE GRAPHS.

To generate embeddings from the constructed KG, this study employs Node2Vec architecture as shown in Fig. 7, a state-of-the-art graph embedding algorithm. Node2Vec is particularly effective because it strikes a balance between preserving both homophily (similar nodes) and structural equivalence (nodes playing similar roles) [28].

How Node2Vec Works:

1. Biased Random Walks:
   Node2Vec simulates a series of random walks across the graph starting from each node. Unlike standard random walks, Node2Vec introduces two parameters—p (return parameter) and q (in-out parameter)—which control the breadth-first (BFS) and depth-first (DFS) search behavior:

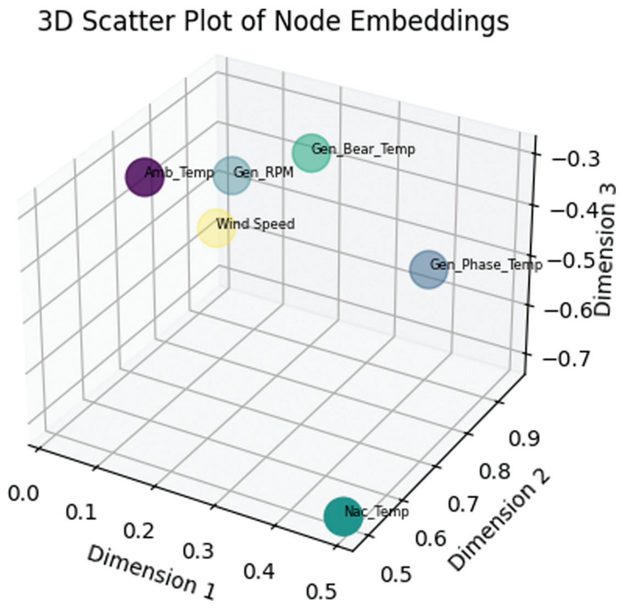


Fig. 8. 3D scatter plot of Node2Vec embeddings for each SCADA parameter node in the knowledge graph. The three axes (dimension 1, 2, and 3) represent latent vector space dimensions learned through biased random walks. Proximity between points (e.g., Gen_Phase_Temp and Gen_Bear_Temp) indicates topological and functional similarity. These embeddings were combined with original SCADA features to build the augmented input space for ML modeling.

- ○ High p, low q → encourages exploring nearby nodes (local context).
- ○ Low p, high q → encourages exploring distant nodes (global context).

2. Context Generation:
   These walks produce sequences of nodes that are treated like “sentences” in natural language processing.
3. Embedding with Skip-Gram Model:
   The sequences are then fed into a Skip-Gram model (like Word2Vec) to learn vector representations. The idea is that nodes that appear in similar walks (contexts) should have similar embeddings.
4. Output:
   Each node (i.e., each SCADA parameter) is assigned a d-dimensional vector that encodes its structural position and relational role in the graph.

2) VISUALIZATION OF NODE EMBEDDINGS. To interpret the semantic and structural insights captured by the Node2Vec embedding process, a 3D scatter plot of the generated node embeddings has been illustrated in Fig. 8. This plot represents a visual distribution of the selected SCADA parameters in a reduced three-dimensional space, with each axis corresponding to one of the embedding dimensions. Each point in the 3D space corresponds to a node in the KG, i.e., a sensor parameter such as Ambient Temperature (Amb_Temp), Wind Speed, Generator RPM (Gen_RPM), or Generator Bearing Temperature (Gen_ Bear_Temp). The spatial proximity between the nodes reflects the contextual similarity learned by Node2Vec during biased random walks on the graph. Clusters of nodes (e.g., Gen_Bear_Temp and Gen_Phase_Temp) indicate strong relational or topological similarity, suggesting they may influence each other in operational scenarios or share similar roles in fault propagation. Conversely, parameters like Nac_Temp are more distantly positioned, indicating weaker or indirect relationships with other nodes in the context of generator bearing failure.

G. ML MODEL TRAINING ON AUGMENTED DATASET

Once the SCADA parameters and their graph-based embeddings are generated using Node2Vec, the next step involves creating an augmented dataset by concatenating the original feature values with the corresponding node embeddings. This fusion of sensor data and semantic information leads to a richer representation of the system, enabling the ML model to learn not just from raw sensor readings but also from the latent interdependencies and contextual knowledge captured via the KG.
Structure of the augmented dataset
Each instance in the augmented dataset now includes:

- Original time-series features: e.g., wind speed, ambient temperature, generator RPM, etc.
- Embedding dimensions: learned 3D node embeddings representing each parameter’s relationship in the graph.

A sample of the augmented dataset is shown in Fig. 9, where embedding vectors (outlined in red) are appended to each time-series observation (outlined in green). This enriched data structure allows the model to leverage domain relationships and contextual cues alongside numerical trends.

The augmented dataset was used to train an XGboost ML model, and then its performance was compared with non-embedded dataset across different metrics like MAE, mean square error (MSE), RMSE, R-Square, and Fit Time. The results for the same are presented and discussed in next section.

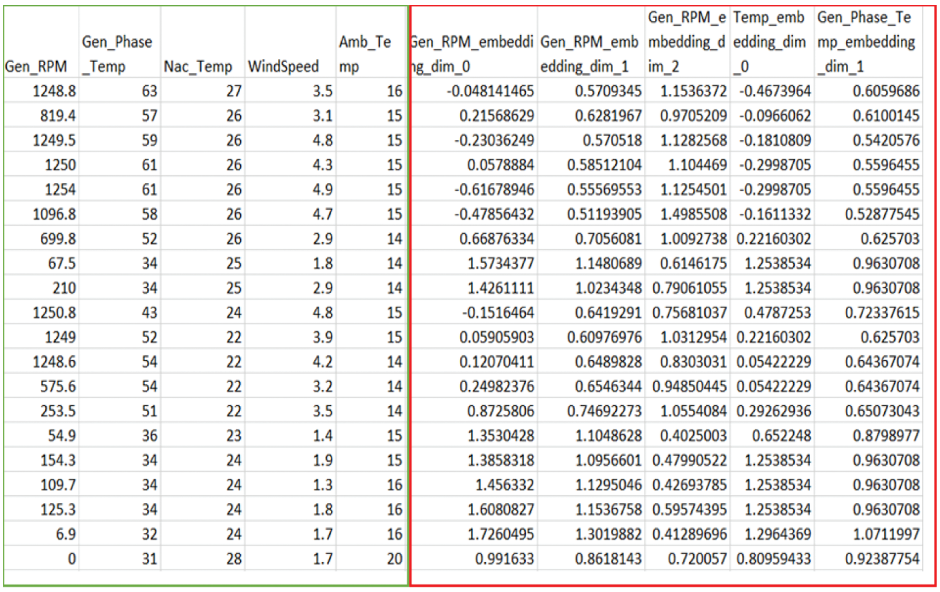
IV. RESULTS AND DISCUSSION

A. PERFORMANCE COMPARISON:

To evaluate the performance of the ML model trained on the augmented dataset (KG + ML) versus the baseline model trained on raw sensor data alone (ML), multiple evaluation metrics were used, including:

- Mean absolute error (MAE)
- Mean squared error (MSE)
- Root mean squared error (RMSE)
- R-Squared (R^2)
- Fit Time (in seconds)

The results of performance metrics, as shown in Fig. 10 between ML and KG+ML models, prove how the incorporation of KG into ML processes delivers substantial advantages. The KG+ML model demonstrates superior accuracy during prediction by combining quick convergence with lower final error compared to the single ML model based on RMSE assessments. Analysis through MAE curves demonstrates that the KG+ML model consistently produced smaller average errors, which signifies exact predictions during the whole training process. The R^2 plots demonstrate



Gen_RPM	Gen_Phase_Temp	Nac_Temp	WindSpeed	Amb_Temp	Gen_RPM_embedding_dim_0	Gen_RPM_embedding_dim_1	Gen_RPM_embedding_dim_2	Temp_embedding_dim_0	Gen_Phase_Temp_embedding_dim_1
1248.8	63	27	3.5	16	-0.048141465	0.5709345	1.1536372	-0.4673964	0.6059686
819.4	57	26	3.1	15	0.21568629	0.6281967	0.9705209	-0.0966062	0.6100145
1249.5	59	26	4.8	15	-0.23036249	0.570518	1.1282568	-0.1810809	0.5420576
1250	61	26	4.3	15	0.0578884	0.58512104	1.104469	-0.2998705	0.5596455
1254	61	26	4.9	15	-0.61678946	0.55569553	1.1254501	-0.2998705	0.5596455
1096.8	58	26	4.7	15	-0.47856432	0.51193905	1.4985508	-0.1611332	0.52877545
699.8	52	26	2.9	14	0.66876334	0.7056081	1.0092738	0.22160302	0.625703
67.5	34	25	1.8	14	1.5734377	1.1480689	0.6146175	1.2538534	0.9630708
210	34	25	2.9	14	1.4261111	1.0234348	0.79061055	1.2538534	0.9630708
1250.8	43	24	4.8	15	-0.1516464	0.6419291	0.75681037	0.4787253	0.72337615
1249	52	22	3.9	15	0.05905903	0.60976976	1.0312954	0.22160302	0.625703
1248.6	54	22	4.2	14	0.12070411	0.6489828	0.8303031	0.05422229	0.64367074
575.6	54	22	3.2	14	0.24982376	0.6546344	0.94850445	0.05422229	0.64367074
253.5	51	22	3.5	14	0.8725806	0.74692273	1.0554084	0.29262936	0.65073043
54.9	36	23	1.4	15	1.3530428	1.1048628	0.4025003	0.652248	0.8798977
154.3	34	24	1.9	15	1.3858318	1.0956601	0.47990522	1.2538534	0.9630708
109.7	34	24	1.3	16	1.456332	1.1295046	0.42693785	1.2538534	0.9630708
125.3	34	24	1.8	16	1.6080827	1.1536758	0.59574395	1.2538534	0.9630708
6.9	32	24	1.7	16	1.7260495	1.3019882	0.41289696	1.2964369	1.0711997
0	31	28	1.7	20	0.991633	0.8618143	0.720057	0.80959433	0.92387754

Fig. 9. A sample of the augmented dataset showing combined original SCADA features (outlined in green) and Node2Vec embeddings (outlined in red) used for model training.

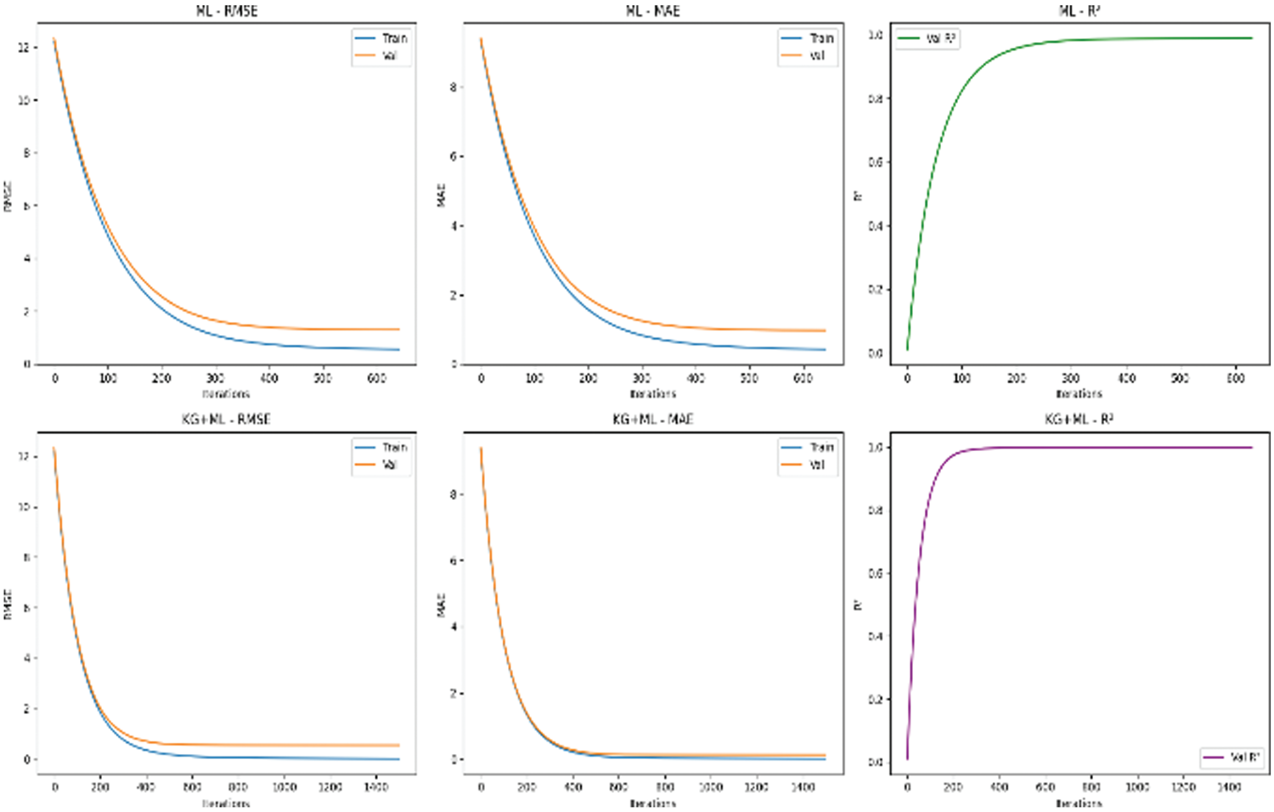


Fig. 10. Training and validation performance comparison of ML and KG + ML models across RMSE, MAE, and R^2 metrics.

that the KG+ML approach reaches perfect explanatory power quickly before the ML model does. The KG+ML model exceeds the ML model by showing better generalization ability throughout its initial stages of operation. The learning process and data representation benefit from the semantic structure and contextual relationships of the KG while guiding the acquisition of better data representations. The KG+ML model provides better results than standard ML approaches in all evaluation indicators, which proves its superior efficiency and reliability. Real-world applications stand to gain from performing domain modeling with data-driven models as an integrated system for better performance. KG proves to be an important tool for enhancing both learning process performance and educational results.

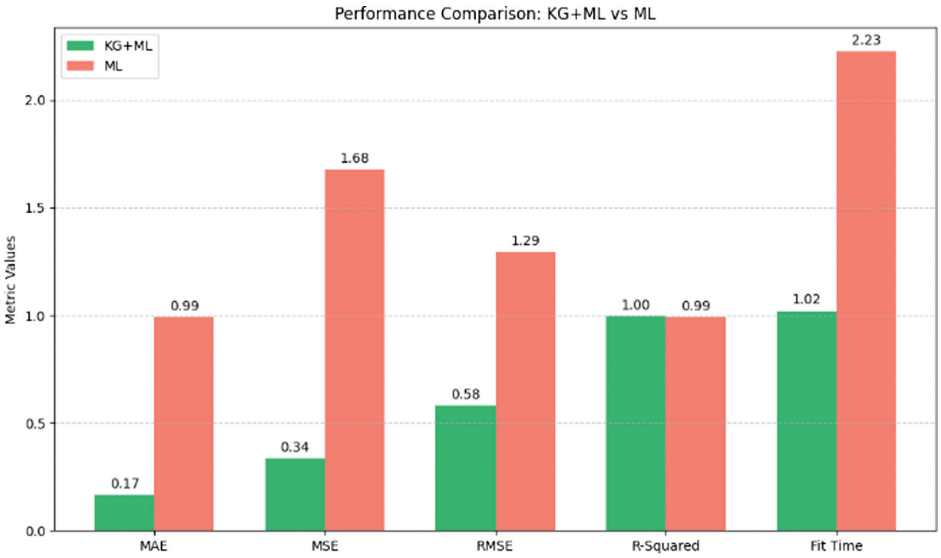


Fig. 11. Performance comparison between ML and KG + ML models across various metrics.

As shown in Fig. 11, the KG + ML model significantly outperforms the traditional ML model across all accuracy metrics:

These results clearly highlight the value addition of KG-based embeddings in enhancing model understanding of contextual dependencies among SCADA parameters. The KG + ML model achieves lower errors, better generalization (as evident from the high R^2 score), and faster training times. With XGBoost, we tested the efficacy of other common ML models like RF and SVM with both ML and KG + ML settings. Although XGBoost outperformed all other models on every measure overall, all models bore substantial gains upon the application of KG embeddings. This proves that the performance gains recorded can be attributed to the KG integration but not to the model-specific benefits. The MAE, RMSE, and R^2 scores of the various models are listed in detail in Appendix A.

B. STATISTICAL ANALYSIS AND RESULTS

A paired sample t-test examined the performance evaluation between KG + ML and conventional ML by analyzing five key measures, including MAE, MSE, RMSE, R^2 score, and Fit Time. The assessment metrics focused on accuracy measures and efficiency assessment aspects of the predictions.

Hypothesis Formulation

These are the hypotheses that define the required test:
Let's define:

- x_i: performance of KG + ML
- y_i: performance of ML
- $d_i = x_i - y_i$: paired difference

Null Hypothesis (H_o): There is no significant difference between KG + ML and ML standalone performance across different metrics.

$$H_o : \mu_d = 0$$

Where μ_d is the mean of the paired difference ($d_i = x_i - y_i$).

Alternate Hypothesis (H_1): There is a significant difference in the performance of the KG + ML and ML models. $H_1 : \mu_d \neq 0$

Paired t-test calculation

Here's a detailed breakdown of the statistical significance testing between the KG + ML model and the ML model across five performance metrics (MAE, MSE, RMSE, R^2, and Fit Time):

Step 1: Compute differences

$d_1 = 0.167 – 0.991 = 0.824$
$d_2 = 0.336 – 1.676 = -1.340$
$d_3 = 0.580 – 1.294 = -0.714$
$d_4 = 0.997 – 0.989 = +0.008$
$d_5 = 1.020 – 2.225 = -1.205$

Step 2: Mean and standard deviation of differences

$$\bar{d} = \frac{-0.824 - 1.340 - 0.714 - 0.008 - 1.205}{5}$$

$$= \frac{-4.075}{5} = -0.815$$

$$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(di - \bar{d})\hat{2}}$$

$$s_d = \sqrt{\frac{1.112696}{4}} = \sqrt{0.278174} = 0.5274$$

Step 3: t-statistic

$$t = \frac{\bar{\boldsymbol{d}}}{\frac{s_d}{\sqrt{n}}} = -3.454$$

Degrees of freedom = 4 (Using a t-distribution table)

p-value ≈ 0.026 (two-tailed)

Since the p-value is less than 0.05, we reject the null hypothesis at the 5% significance level. This indicates that the difference in performance between the KG + ML and ML models is statistically significant and not due to random chance.

C. INTERPRETABILITY COMPARISONS OF KG + ML OVER ML

Table II presents the interpretability of feature importance scores generated by traditional ML methods. Traditional ML calculates feature importance by examining feature contributions to error reduction, yet this method captures only basic statistical relationships [29]. The technique fails to reveal the complete structural relationships and underlying causations, which exist between different variables. Features in the KG maintain interconnected entity relationships, which allow for the calculation of centrality metrics [30] through measures of degree centrality, in-degree, out-degree, and closeness centrality as shown in Table III. System structure metrics show a feature's degree of influence through the network to identify its position as either fault cause, intermediary, or final fault outcome. The centrality analysis demonstrates that Wind Speed, together with Rotor RPM, stands out as key system component, while their ML importance scores are only moderate. Traditional ML techniques fail to detect the type of structural understanding, which this method provides. A combination of statistical learning and graph-based reasoning through the KG + ML approach delivers an enhanced framework, which effectively predicts faults and establishes clear explanations for what data points matter to the WT system.

This analysis shows the KG + ML approach gives better interpretability through its KG-based centrality measures compared to XGBoost-based traditional ML modeling methods. The XGBoost model identifies Gen_Phase_Temp as its most influential feature with a score of 8.47 and Nac_Temp follows with a score of 2.23, while Amb_Temp receives a score of 0.80. Each feature rating reveals its predictive strength, which depends mostly on data variability and split performance. Through its KG structure, the KG applies three measures of graph centrality—degree centrality and in/out-degree and closeness centrality—to determine the contextual and relational value of features based on domain expertise. The two features, Gen_Phase_Temp and Nac_Temp, demonstrate high degree centrality ratings (0.8) because they connect to a large number of interactions that occur throughout the turbine system. Gen_Phase_Temp functions as an essential feature in tracking system-level dynamics since it receives influence through three distinct upstream features, as shown by its in-degree score of 3. The ML model attributes low importance to Amb_Temp and WindSpeed but these variables demonstrate three out-degree links in the KG. The KG indicates that these factors serve as important initial nodes influencing multiple follow-up factors, which traditional predictive models could miss because they mainly detect direct target correlations. By integrating both KG + ML techniques, the approach reaches superior prediction results while delivering more meaningful domain-based explanations. This methodology features relational dependencies and causal pathways to enable stakeholders better understanding of system behavior and feature interactions, specifically needed when detecting faults in safety-critical applications like WTs. The experimental findings demonstrate the significant potential of integrating KG embeddings into ML workflows. The graph-based approach captures complex inter-variable dependencies, enabling the model to learn richer representations compared to conventional ML that treats each feature independently. The Node2Vec algorithm, used for graph embeddings, facilitated the transformation of nodes (SCADA parameters) into low-dimensional vectors that reflect their structural and contextual similarities in the KG. These embeddings were then appended to the original dataset, forming an augmented dataset that improved model accuracy, reduced training error, and decreased computational time. This implies that structural domain knowledge, when encoded into embeddings, complements sensor data effectively and contributes to more robust and interpretable ML models in WT monitoring system. Furthermore, the t-test supports the claim that KG + ML significantly outperforms ML.

Table II. ML feature importance (XGBoost)

Feature	Feature importance
Gen_Phase_Temp	8.47
Nac_Temp	2.23
Amb_Temp	0.800
Gen_RPM	0.598
WindSpeed	0.238

V. CONCLUSION

This work introduces a novel method that combines KG with conventional ML techniques in order to improve predictive performance for WT anomaly detection. The enhanced dataset produced by utilizing the domain knowledge recorded in a KG was more informative and richer in semantics than the raw sensor data by itself. The outcomes show that the KG + ML strategy outperforms solo ML techniques in a number of performance parameters, such as Fit Time, MAE, MSE, RMSE, and R^2. Notably, the KG + ML model enhanced model fitting efficiency and drastically reduced prediction error (MAE: 0.167 vs. 0.991; RMSE: 0.58 vs. 1.294). Furthermore, a statistical study employing a paired t-test validated that the performance gains were not the result of random fluctuation and

Table III. Contextual feature importance from graph

Node	Degree centrality	In-degree	Out-degree	Closeness centrality
Gen_Phase_Temp	0.8	3	1	0.64
Nac_Temp	0.8	2	2	0.40
Amb_Temp	0.6	0	3	0.00
WindSpeed	0.6	0	3	0.00
Gen_RPM	0.6	1	2	0.20
Gen_bearing_temp	1.0	5	0	1.00

established the significance ($p = 0.026$) of these improvements. Importantly, the interpretability of the model was enhanced through centrality-based feature relevance analysis, offering insights into causal and structural relationships that standard feature importance techniques may miss. By incorporating contextual linkages that traditional ML alone was unable to grasp, the augmented dataset acted as a crucial bridge. This dataset's extended feature space was demonstrated by a sample, which also revealed new insights from domain-specific limitations, operational dependencies, and turbine topology. In summary, this research underscores the potential of KG-enhanced learning pipelines in industrial monitoring systems. By embedding expert knowledge into the training process, we bridge the gap between data-driven models and real-world system understanding, leading to more reliable and interpretable results. Future work may explore real-time deployment, integration with digital twins, and application across other critical infrastructure systems.

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest.

REFERENCES

[1] R. S. S. Nuvvula et al., "Machine Learning-Driven Predictive Maintenance Framework for Anomaly Detection and Prognostics in Wind Farm Operations," in *2024 12th International Conference on Smart Grid (icSmartGrid)*, IEEE, May 2024, pp. 284–289. doi: 10.1109/icSmartGrid61824.2024.10578083.

[2] C. Zhu and Y. Li, "Reliability Analysis of Wind Turbines," in Stability Control and Reliable Performance of Wind Turbines, London, UK: InTech, 2018, pp. 169–186. doi: 10.5772/intechopen.74859.

[3] Shuai Shi and K. L. Lo, "An overview of wind energy development and associated power system reliability evaluation methods," in *2013 48th International Universities' Power Engineering Conference (UPEC)*, IEEE, Sep. 2013, pp. 1–6. doi: 10.1109/UPEC.2013.6714894.

[4] C. Dao, B. Kazemtabrizi, and C. Crabtree, "Wind turbine reliability data review and impacts on levelised cost of energy," *Wind Energy*, vol. 22, no. 12, pp. 1848–1871, Dec. 2019, doi: 10.1002/we.2404.

[5] T. S. Selwyn and R. Kesavan, "Computation of reliability and birnbaum importance of components of a wind turbine at high uncertain wind.," *Int J Comput Appl.*, vol. 975, pp. 8887, 2011.

[6] E. Hart et al., "A review of wind turbine main bearings: design, operation, modelling, damage mechanisms and fault detection," *Wind Energy Science.*, vol. 5, no. 1, pp. 105–124, Jan. 2020, doi: 10.5194/wes-5-105-2020.

[7] A. Dhanola and H. C. Garg, "Tribological challenges and advancements in wind turbine bearings: A review," *Eng Fail Anal*, vol. 118, pp. 104885, Dec. 2020, doi: 10.1016/j.engfailanal.2020.104885.

[8] M. J. Roemer and C. S. Byington, "Prognostics and health management software for gas turbine engine bearings," in Volume 1: Turbo Expo 2007, New York, NY, USA: ASMEDC, Jan. 2007, pp. 795–802. doi: 10.1115/GT2007-27984.

[9] R. F. Orsagh, J. Sheldon, and C. J. Klenke, "Prognostics/diagnostics for gas turbine engine bearings," in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, IEEE, pp. 3095–3103. doi: 10.1109/AERO.2003.1234152.

[10] J.-Y. Hsu et al., "Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning," *IEEE Access*, vol. 8, pp. 23427–23439, 2020, doi: 10.1109/ACCESS.2020.2968615.

[11] M. Garan, K. Tidriri, and I. Kovalenko, "A data-centric machine learning methodology: Application on predictive maintenance of wind turbines," *Energies (Basel)*, vol. 15, no. 3, pp. 826, Jan. 2022, doi: 10.3390/en15030826.

[12] C.-H. Yeh et al., "Machine Learning for Long Cycle Maintenance Prediction of Wind Turbine," *Sensors*, vol. 19, no. 7, p. 1671, Apr. 2019, doi: 10.3390/s19071671.

[13] X. Shu and Y. Ye, "Knowledge discovery: Methods from data mining and machine learning," *Soc Sci Res.*, vol. 110, pp. 102817, Feb. 2023, doi: 10.1016/j.ssresearch.2022.102817.

[14] M. Cheng et al., "Automated knowledge graphs for complex systems (AutoGraCS): Applications to management of bridge networks," *Resilient Cities and Structures*, vol. 3, no. 4, pp. 95–106, Dec. 2024, doi: 10.1016/j.rcns.2024.11.001.

[15] R. Ghiasi et al., "An unsupervised anomaly detection framework for onboard monitoring of railway track geometrical defects using one-class support vector machine," *Eng Appl Artif Intell.*, vol. 133, pp. 108167, Jul. 2024, doi: 10.1016/j.engappai.2024.108167.

[16] Z. Soltani, H. Hassani, and S. Esmaeiloghli, "A deep autoencoder network connected to geographical random forest for spatially aware geochemical anomaly detection," *Comput Geosci.*, vol. 190, pp. 105657, Aug. 2024, doi: 10.1016/j.cageo.2024.105657.

[17] Y. Wang et al., "IDS-KG: An industrial dataspace-based knowledge graph construction approach for smart maintenance," *J Ind Inf Integr.*, vol. 38, pp. 100566, Mar. 2024, doi: 10.1016/j.jii.2024.100566.

[18] Q. Yang et al., "A semantic Enhanced Knowledge Graph Embedding Model With AIGC Designed for Healthcare Prediction," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2024, doi: 10.1109/TCE.2024.3398207.

[19] W. Zhang et al., "Billion-scale Pre-trained E-commerce Product Knowledge Graph Model," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, IEEE, Apr. 2021, pp. 2476–2487. doi: 10.1109/ICDE51399.2021.00280.

[20] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, and Y. Zhou, "Recent Progress of Using Knowledge Graph for Cybersecurity," *Electronics (Basel)*, vol. 11, no. 15, p. 2287, Jul. 2022, doi: 10.3390/electronics11152287.

[21] Q. Bao et al., "A node2vec-based graph embedding approach for unified assembly process information modeling and workstep execution time prediction," *Comput Ind Eng.*, vol. 163, pp. 107864, Jan. 2022, doi: 10.1016/j.cie.2021.107864.

[22] G. V. R. Babu et al., "Dynamic Fault Diagnosis in Wind Turbines: A GNN-LSTM Approach," in *2024 IEEE 3rd International Conference on Electrical Power and Energy Systems (ICEPES)*, IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ICEPES60647.2024.10653514.

[23] M. Nickel et al., "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE.*, vol. 104, no. 1, pp. 11–33, Jan. 2016, doi: 10.1109/JPROC.2015.2483592.

[24] S. Pan et al., "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans Knowl Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024, doi: 10.1109/TKDE.2024.3352100.

[25] B. Jia et al., “Pattern Discovery and Anomaly Detection via Knowledge Graph,” in *2018 21st International Conference on Information Fusion (FUSION)*, IEEE, Jul. 2018, pp. 2392–2399. doi: 10.23919/ICIF.2018.8455737.

[26] A. Liu et al., “Knowledge graph with machine learning for product design,” *CIRP Annals*., vol. 71, no. 1, pp. 117–120, 2022, doi: 10.1016/j.cirp.2022.03.025.

[27] O. T. Bindingsbø et al., “Fault detection of a wind turbine generator bearing using interpretable machine learning,” *Front Energy Res*., vol. 11, Dec. 2023, p. 1284676. doi: 10.3389/fenrg.2023.1284676.

[28] M. Grohe, “Word2vec, node2vec, graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data,” in Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, New York, NY, USA, ACM, Jun. 2020, pp. 1–16. doi: 10.1145/3375395.3387641.

[29] G. K. Rajbahadur et al., “The impact of feature importance methods on the interpretation of defect classifiers,” *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2245–2261, Jul. 2022, doi: 10.1109/TSE.2021.3056941.

[30] M. Nunes et al., “Creating actionable and insightful knowledge applying graph-centrality metrics to measure project collaborative performance,” *Sustainability*, vol. 14, no. 8, pp. 4592, Apr. 2022, doi: 10.3390/su14084592.

Appendix A

Performance comparison of ML and KG+ML models

Model	Configuration	MAE	MSE	RMSE	R^2
XGBoost	**ML**	**0.991**	**1.68**	**1.294**	**0.989**
XGBoost	**KG + ML**	**0.167**	**0.34**	**0.580**	**0.999**
Random Forest	**ML**	**1.102**	**1.78**	**1.382**	**0.981**
Random Forest	**KG + ML**	**0.221**	**0.56**	**0.642**	**0.993**
SVM	**ML**	**1.234**	**1.98**	**1.412**	**0.975**
SVM	**KG + ML**	**0.283**	**0.65**	**0.691**	**0.989**